10/081,061

日本国特許庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出願年月日
Date of Application: ２００１年 ４月２４日

出願番号
Application Number: 特願２００１－１２５２２４

[ST.10/C]: [ＪＰ２００１－１２５２２４]

出願人
Applicant(s): 株式会社日立製作所

CERTIFIED COPY OF
PRIORITY DOCUMENT

２００２年 ２月１５日

特許庁長官
Commissioner,
Japan Patent Office 及川耕造

出証番号 出証特２００２－３００７５８９

【書類名】　特許願

【整理番号】　K01007871A

【あて先】　特許庁長官

【国際特許分類】　G06F　9/46

【発明者】

【住所又は居所】　神奈川県川崎市麻生区王禅寺１０９９番地　株式会社日立製作所　システム開発研究所内

【氏名】　宮崎　邦彦

【発明者】

【住所又は居所】　神奈川県川崎市麻生区王禅寺１０９９番地　株式会社日立製作所　システム開発研究所内

【氏名】　伊藤　信治

【発明者】

【住所又は居所】　神奈川県川崎市麻生区王禅寺１０９９番地　株式会社日立製作所　システム開発研究所内

【氏名】　吉浦　　裕

【発明者】

【住所又は居所】　神奈川県川崎市麻生区王禅寺１０９９番地　株式会社日立製作所　システム開発研究所内

【氏名】　宝木　和夫

【発明者】

【住所又は居所】　神奈川県川崎市麻生区王禅寺１０９９番地　株式会社日立製作所　システム開発研究所内

【氏名】　荒井　正人

【発明者】

【住所又は居所】　神奈川県川崎市麻生区王禅寺１０９９番地　株式会社日立製作所　システム開発研究所内

【氏名】　新井　利明

【発明者】

【住所又は居所】　神奈川県川崎市幸区鹿島田８９０番地　株式会社日立製作所　情報サービス事業部内

【氏名】　松木　　武

【発明者】

【住所又は居所】　東京都江東区新砂一丁目６番２７号　株式会社日立製作所　公共システム事業部内

【氏名】　豊島　　久

【特許出願人】

【識別番号】　000005108

【氏名又は名称】　株式会社日立製作所

【代理人】

【識別番号】　100075096

【弁理士】

【氏名又は名称】　作田　康夫

【手数料の表示】

【予納台帳番号】　013088

【納付金額】　21,000円

【提出物件の目録】

【物件名】　明細書　1

【物件名】　図面　1

【物件名】　要約書　1

【プルーフの要否】　要

【書類名】　明細書

【発明の名称】　高信頼計算機システム

【特許請求の範囲】

【請求項１】ファイルアクセスを実行する入出力処理部と、

ユーザの指示により、前記入出力処理部へファイルアクセスを要求するアクセス実行処理部と、

前記ファイルアクセス実行時にアクセス制御を行うアクセス制御部を備える計算機システムであって、

前記アクセス制御部は、

前記アクセス実行処理部から保護された記憶部と、

前記ファイルのセキュリティレベルを記述した、前記記憶部中のファイルリストと、

前記ユーザのクリアランスを記述した、前記記憶部中のユーザリストと、

前記ファイルアクセスの正当性を、前記ファイルリストと、前記ユーザリストと、前記ファイルアクセスのアクセスタイプと、前記ファイルを特定する情報と、前記ユーザを特定する情報と、に基づき判定するアクセス制御処理部と、

前記入出力処理部によるファイルアクセスの実行時に、前記アクセスタイプと、前記ファイルを特定する情報と、前記ユーザを特定する情報と、を前記アクセス制御処理部に送り、

前記ファイルアクセスの正当性判定結果を前記アクセス制御処理部から受け取り、

前記ファイルアクセスが正当であれば、前記入出力処理部によるファイルアクセスを実行させ、不当であれば前記ファイルアクセスを中止させるアクセス監視処理部と、を備える

アクセス制御部を備える計算機システム。

【請求項２】請求項１に記載のアクセス制御部を備える計算機システムであって、

前記アクセス制御処理部が使用する前記記憶部中の記憶領域を、前記アクセス実行処理部から保護するための排他制御部を備える

アクセス制御部を備える計算機システム。

【請求項３】請求項２記載のアクセス制御部を備える計算機システムであって、

前記ユーザリストの設定と管理を行うためのユーザリスト設定管理処理部を備える

アクセス制御部を備える計算機システム。

【請求項４】請求項３記載のアクセス制御部を備える計算機システムであって、

前記ユーザリスト設定処理部は、セキュリティ管理者の認証処理部を備える

アクセス制御部を備える計算機システム。

【請求項５】請求項４記載のアクセス制御部を備える計算機システムであって、

前記セキュリティ管理者は、前記アクセス実行処理部を管理するシステム管理者とは異なる

アクセス制御部を備える計算機システム。

【請求項６】請求項１記載のアクセス制御部を備える計算機システムであって、

前記ファイルリストの設定と管理を行うためのファイルリスト設定管理処理部を備える

アクセス制御部を備える計算機システム。

【請求項７】請求項６記載のアクセス制御部を備える計算機システムであって、

前記ファイルリスト設定管理処理部は、セキュリティ管理者の認証処理部を備える

アクセス制御部を備える計算機システム。

【請求項８】請求項７記載のアクセス制御部を備える計算機システムであって、

前記セキュリティ管理者は、前記アクセス実行処理部を管理するシステム管理者とは異なることを特徴とする、

アクセス制御部を備える計算機システム。

【請求項９】請求項１記載のアクセス制御部を備える計算機システムであって、

記憶媒体へファイルの出力を要求するファイルアクセスが正当であれば、前記ファイルを暗号化する暗号処理部と、

記憶媒体から暗号化されたファイルの入力を要求するファイルアクセスが正当であれば、前記暗号化されたファイルを復号化する復号処理部を備える

アクセス制御部を備える計算機システム。

【請求項１０】請求項９記載のアクセス制御部を備える計算機システムであって、

前記排他制御部は、前記暗号処理部と復号処理部で利用される、少なくともひとつの鍵情報が記憶される前記記憶部中の記憶領域を前記アクセス実行処理部から保護する

アクセス制御部を備える計算機システム。

【請求項１１】請求項９記載のアクセス制御部を備える計算機システムであって、

前記暗号処理部と復号処理処理部は、前記ファイルリストに記述されたセキュリティレベル毎に異なる、複数の鍵情報と少なくとも一つの暗号方法を利用するアクセス制御部を備える計算機システム。

【請求項１２】請求項１記載のアクセス制御部を備える計算機システムであって、

前記入出力処理部またはアクセス監視処理部が、改ざんされていないこと、あるいはあらかじめ定められた動作をしていることを監査し、前記入出力処理部またはアクセス監視処理部が改ざんされている場合、あるいはあらかじめ定められた動作と異なる動作をしている場合には、ファイルの入出力を中止を指示する入出力処理監査処理部を備える

アクセス制御部を備える計算機システム。

【請求項１３】請求項１記載のアクセス制御部を備える計算機システムであって、

前記アクセス制御処理部に伝えられたファイルアクセスに関する情報を記録し、管理するファイルアクセスログ処理部を備える

アクセス制御部を備える計算機システム。

【請求項１４】請求項１記載のアクセス制御部を備える計算機システムであって、

前記アクセス制御部は、ソフトウェアモジュールとして実現されているアクセス制御部を備える計算機システム。

【請求項１５】請求項１記載のアクセス制御部を備える計算機システムであって、

前記アクセス制御部は、ハードウェアモジュールとして実現されているアクセス制御部を備える計算機システム。

【請求項１６】各種処理に必要な情報を格納するための記憶部と、

暗号の復号化処理部、または、ディジタル署名の生成処理部を含む、暗号機能処理部と、

前記暗号機能処理部で利用される鍵情報を保持する鍵記憶部と、

ユーザの指示により、暗号機能実行を要求する暗号利用要求実行処理部と、

前記暗号利用要求実行処理部から前記暗号機能処理部に対する暗号機能実行要求を、監視して前記暗号機能処理部に伝え、処理結果を前記暗号機能処理部から受信する暗号利用要求監視処理部と、

前記暗号機能処理部が使用する前記記憶部中の記憶領域、または、鍵記憶部を、前記暗号利用要求実行処理部から保護するための排他制御部とを備える
計算機システム。

【請求項１７】請求項１６記載の計算機システムであって、

前記暗号機能処理部は、前記ユーザの認証処理部を備える

計算機システム。

【請求項１８】請求項１６記載の計算機システムであって、

前記暗号機能処理部は、ソフトウェアモジュールとして実現されている計算機システム。

【請求項１９】請求項１６記載の計算機システムであって、

前記暗号機能処理部は、ハードウェアモジュールとして実現されている計算機システム。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】本発明は情報処理システムに保管されているファイルへの不正アクセスを防止し、高い安全性を確保するための技術に関する。

【０００２】

【従来の技術】計算機を用いた情報処理システムにおいては、多くのアプリケーションプログラムを実行させるための基本的な仕組みを提供するOS（Oprerating System)を用いてハードウェア資源の有効活用を図るのが一般的である。

【０００３】

さらに、当該情報処理システムでのファイルの保護は、主として、上記OSが備える一機能であるファイルアクセス制御機能を用いて実現している。
現在、広範に利用されているOSでは、ファイルアクセス制御を、任意アクセス制御(Discretionary Access Control、DACとも記す)に基づいて行うことが多い。これは、ファイルの所有者(一般には、ファイルの作成者と一致する)であるユーザが、当該ファイルに対して、アクセスできるユーザ、および、アクセスの種類（読み取りのみ、読み取りと変更など）を任意に設定し、それに基づいてアクセス制御を行うものである。

また、通常このようなOSでは、システム管理者と呼ばれる特殊な権限をもつユーザは、当該情報処理システム内のすべての設定を変更することが可能である。

【０００４】

一方、高いセキュリティが要求される場面での利用を想定して作られた専用OSの中には、より厳格なアクセス制御方法である、強制アクセス制御(必須アクセス制御とも言う)(Mandatory Access Control、MACとも記す)に基づくファイルアクセス制御機能を備えたものがある。たとえば、セキュアシステムの評価基準のひとつである米国のTCSEC(Trusted ComputerSystem Evaluation Criteria)に定められたクラスのうち、クラスB1以上に認定されるためには、強制アクセス制御機能を備えなければならない。

TCSECおよび強制アクセス制御については、
文献[TCSEC]：「Department of Defense Trusted Computer System Evaluation Criteria」アメリカ国防総省標準（DOD 5200.28-STD)
に詳しい。

【０００５】

強制アクセス制御を備えた情報処理システムにおいては、すべてのファイルには、セキュリティレベル(例えば、高い方から「機密（top secret)」「極秘（secre

t)」「秘密（confidential)」「一般（unclassified)」など)が設定され、また、すべてのユーザに対し、信頼度、すなわち、アクセスできるファイルのセキュリティレベルを示す許可レベル(クリアランス(clearance)という。例えば、高い方から「機密」「極秘」「秘密」「一般」など))が設定される。システムは、これらの情報に基づいて強制的にアクセス制御を行う。

【０００６】

上記のような専用OSでは、システム管理者とは異なる、セキュリティ管理者とよばれるユーザによって、ファイルのセキュリティレベルや他のユーザのクリアランスなど、セキュリティに関わる機能が管理されていることが多い。これにより、ファイル作成者だけでなくシステム管理者であっても当該情報処理システムのセキュリティを下げることができないようになり、より安全なシステムの運用が可能となる。

【０００７】

【発明が解決しようとする課題】

ファイルへの不正アクセスを防ぐ仕組みを任意アクセス制御に基づくアクセス制御機能を備えたOSに依存している情報処理システムでは、各ファイルの所有者であるユーザが、当該ファイルのアクセス可能な範囲を設定できてしまうため、ユーザの故意、または、不注意により、重要な情報が漏洩してしまう恐れがある。

【０００８】

一方、強制アクセス制御機能を備えたOSを利用した情報処理システムでは、当該OSが一般的ではないため、ユーザが利用するアプリケーションプログラムを独自に開発する必要があり、コストがかさむ上、ユーザの利便性が失われる。

上記背景の元、コストがかからず、安全性を高められる技術が求められている。

【０００９】

【課題を解決するための手段】

本発明は、情報処理システムの情報セキュリティに関する安全性を、利用者の利便性を損なうことなく、向上させる技術を提供する。

本発明は、また、情報処理システム内にシステム管理者とは異なる、資格を持つユーザだけが利用できる機能・領域を設けるための方法、および、装置を提供する。

本発明は、また、上記技術を適用した情報処理システムに対する、起動時における不正を防止するための方法、および、装置を提供する。

【００１０】

より具体的には、既存のアプリケーションプログラムを利用可能な状態を保ちつつ、情報処理システムに強制アクセス制御機能を付加する技術と、その技術を適用した情報処理システムを提供する。

また、強制アクセス制御機能をもたないOSの管理の元で既存のアプリケーションプログラムが動作する情報処理システムに、強制アクセス制御機能を付加する技術と、その技術を適用した情報処理システムを提供する。

【００１１】

本発明の一態様によれば、ユーザが直接利用する、アプリケーションプログラムを動作させるためのOS(ホストOSという)と、セキュリティ強化のためのOS(ゲストOSという)の２つのOSを、複数OS制御技術を利用することにより動作させる。複数OS制御技術は、たとえば、特開平11-149385号公報に開示されている。

本発明に従えば、上記専用OSを用いることなく、ホストOS上で動作するファイル作成者だけでなく、システム管理者であっても、セキュリティレベルを変えることが非常に困難になる。

【００１２】

より具体的には、アクセス監視処理プログラムはホストOS管理下で行われるファイルアクセスを監視し、ゲストOS上のファイルアクセス制御プログラムは、アクセス監視処理プログラムが検知したファイルアクセスの正当性を、ファイルのセキュリティレベルと、ユーザのクリアランスに従って判断し、不正アクセスを許さない。

また、ホストOSの各種設定を行うシステム管理者とは、別に、セキュリティ管理者を設け、システム管理者であっても、セキュリティレベルを変更できないようにする。システム全体のセキュリティポリシーとして、システム管理者が、セ

キュリティ管理者の権限を有することが許される場合には、これら２つの管理者を同一人が兼任してもよい。

【００１３】

本発明の他の一態様によれば、ゲストOS上に、アクセス制御機能を設けることにより、アクセス制御機能に対する不正操作を防止する。

また、本発明の他の一態様によれば、ホストOS側の多機能化・高機能化等の改良が行われても、強制アクセス制御が可能な環境を維持することを可能とする。

さらに、ユーザのクリアランスやファイルのセキュリティレベルの設定にあたっては、セキュリティ管理者の認証を行うことにより、ホストOS側の管理者による不正の脅威を減らすことができるようになる。

【００１４】

なお、本発明におけるOSとは、ユーザーまたはプログラムからの要求に応じて記憶媒体中のデータ、ファイルへのアクセスを実行する機能と、アクセス要求元のユーザーやプログラムを識別する機能と、排他制御機能により占有する記憶部を有するプログラムモジュールを意味しており、

（１）ファイルアクセスを管理しており、検知が可能である。

（２）アクセス要求元のユーザーを識別できる。

（３）アクセス要求元のアプリケーションプログラムを識別できる。

といった特徴を持つ。したがって、上記特徴を持つものであれば、本発明のOSに含めることが可能である。

【００１５】

【発明の実施の形態】図１は、本実施例による情報処理システムの構成例である。コンピュータ１００１内には、コンピュータ内の各OSや各プログラム（処理部とも言う）を実行するCPU１００３、ホストOS１０１７が管理するメモリA１００５、および、ゲストOS１０１９が管理するメモリB１００７があり、さらに、キーボード１０１１、マウスなどの入力装置、ディスプレイ１００９などの出力装置、ホストOS１０１７の管理下にあるハードディスクA１０１３、ゲストOS１０１９の管理下にあるハードディスクB１０１５などの記憶装置がつながっている。なお、記憶装置は、ハードディスクの他、フラッシュメモリや、EEPROMなど書

き換え可能な不揮発性メモリであってもよい。さらには、ハードウェア的な耐タンパ性を持っているほうが望ましい。

【００１６】

メモリA１００５内には、ホストOS１０１７と、さらにホストOS１０１７管理下で動作する一般のアプリケーションプログラム１０２１の他、本実施例では、ユーザのクリアランスを定義するラベル付けを行うユーザラベル設定プログラム１０２３、各ファイルのセキュリティレベルを定義するラベル付けを行うファイルラベル設定プログラム１０２５が、ハードディスクＡ１０１３からロードされる。また、ホストOS１０１７内には、ファイル管理を行う部分として、I/0マネージャ１０２７、ファイルシステムドライバ１０３１、デバイスドライバ１０３３がある他、本実施例では、さらに、アプリケーションからハードディスクA１０１３内のファイル１０４５へのアクセスをフックする、ファイルアクセスフックプログラム１０２９が設けられる。

【００１７】

ファイルへのアクセスをフックするとは、ハードディスクA１０１３等の記憶媒体内のファイルへのアクセスが発生しようとしていることを、何らかの方法によって検知することである。

ファイルアクセスのフックは、たとえば、あらかじめOSによって用意されたインターフェイス、たとえばファイルシステムドライバ１０３１やデバイスドライバ１０３３などの各種ドライバ用に用意されたインターフェイスを利用することが可能である。

すなわち、本実施例のように、I/0マネージャ１０２７と、ファイルシステムドライバ１０３１との間に、I/0マネージャ１０２７からは、ファイルシステムドライバ１０３１に対して命令するように見え、また逆に、ファイルシステムドライバ１０３１からは、I/0マネージャ１０２７から命令されたかのように見える、ファイルアクセスフックプログラム１０２９を設ければ、ファイルアクセスが発生していることを、I/0マネージャ１０２７による命令によって認識することができる。一般的に、OSには、各種デバイスを統一的に管理し、また、様々な機能拡張に対応することができるように、このようなインターフェイスが用意さ

れているし、さらに、ファイルアクセスフックのための専用のインターフェイスが用意されていることもある。

この他、図示されていないが、ホストOS１０１７は、一般にOSが備える機能を実現する部分、すなわち、プロセス管理、メモリ管理などを担う部分を備える。

【００１８】

メモリB１００７内には、ゲストOS１０１９と、さらにゲストOS１０１９管理下で動作する、アクセス制御プログラム１０３５、ユーザリスト管理プログラム１０３７、ファイルリスト管理プログラム１０３９が、ハードディスクＢ１０１５からロードされる。ゲストOS１０１９の管理下にあるハードディスクB１０１５には、ユーザリスト管理プログラム１０３７が利用するユーザリスト１０４１、ファイルリスト管理プログラム１０３９が利用するファイルリスト１０４３が含まれている。

【００１９】

本実施例においては、ホストOS１０１７上のアプリケーションプログラム１０２１によるファイルアクセスは、ファイルアクセスフックプログラム１０２９によりフックされる。

アプリケーションプログラム１０２１から発行されたファイルアクセスを受けたI/Oマネージャ１０２７は、ファイルシステムドライバ１０３１に対してファイルアクセス命令を発行する。ファイルアクセスフックプログラム１０２９は、ファイルシステムドライバ１０３１の代わりにこの命令を受け取り、そのファイルアクセス命令をファイルシステムドライバ１０３１に渡す前に、複数OS制御プログラム１０４９内のOS間通信プログラム１０５１を用いて、ゲストOS１０１９上のアクセス制御プログラム１０３５に渡す。

【００２０】

アクセス制御プログラム１０３５は、当該ファイルアクセスが正当なものか否かを、ユーザリスト１０４１、および、ファイルリスト１０４３を元に判定し判定結果を返す。ファイルアクセスフックプログラム１０２９は、ファイルアクセスが不当であれば、当該ファイルアクセスを失敗させ、正当であれば要求どおりのファイルアクセス命令をファイルシステムドライバ１０３１に渡す。

【００２１】

本実施例では、アクセス制御プログラム１０３５を、ユーザがログインするホストOS１０１７管理下でなく、ゲストOS１０１９管理下で実行することにより、（１）アクセス制御プログラム１０３５が不正な改変にさらされる脅威が少ない（２）アクセス制御に関するプログラムのバグ等によるセキュリティホールがあるかどうかをチェックすべき範囲を限定できるため、メンテナンスが容易になる（３）OSの多機能化・高機能化などによりホストOS１０１７が改良された場合にも、基本的な枠組みを変えることなく、アクセス制御を行うことができるため、OSの開発コストの削減が可能
などの利点が得られる。

【００２２】

本実施例においては、ファイルのセキュリティレベルとして「一般」、「秘密」、「極秘」、「機密」の４段階あり、この順に機密度合いが高くなる。また、対応するユーザのクリアランスとして、同様に「一般」、「秘密」、「極秘」、「機密」の４段階あり、この順に信頼度が高くなる。ファイルアクセスが正当であるか否かは、ファイルのセキュリティレベルとユーザのクリアランスとに基づいて判断される。

【００２３】

セキュリティレベルを何段階設けるかは、システムごとに自由に設定できる。ただし、ファイルのセキュリティレベルとユーザのクリアランスとは1対1に対応するように設定する。

【００２４】

さらに、それぞれのファイルとユーザは「人事部門」、「営業部門」、「企画部門」、「研究部門」の４つの区画(compartment)に所属させることができる。なお、区画の数はシステムごとに自由に設定できる。特に区画が１つの場合は、区画を設けなくてもよい。

【００２５】

各ユーザは、与えられたクリアランスと区画が同じで、かつ、機密度合いが同じか低いセキュリティレベルのファイルは、読み出し可能であり、与えられたクリアランスと区画が同じで、かつ、機密度合いが同じか高いセキュリティレベル

のファイルは、書き込み、または、作成可能である。

見方を変えると、各ファイルは、与えられたセキュリティレベルと機密度合いが同じか低いクリアランスのユーザによる書き込みまたは作成は可能であり、機密度合いが同じか高いクリアランスのユーザによる読み出しは可能である。

【００２６】

強制アクセス制御においては、「極秘」のクリアランスを持つユーザAは、「一般」または「秘密」または「極秘」のセキュリティレベルのファイルを読むことはできるが、「機密」のセキュリティレベルのファイルを読むことはできない。また、ユーザAは、「機密」または「極秘」のセキュリティレベルのファイルに書き込むはできるが、「一般」または「秘密」のファイルに書き込むことはできない。

【００２７】

このように、読み取りが許可されるファイルのセキュリティレベルと、書き込みが許可されるファイルのセキュリティレベルが異なるのは、情報(ファイル)に設定されたセキュリティレベルが下げられてしまうことを防ぐためである。仮に、ユーザAが「一般」のファイルに書き込むことができてしまうとすると、もともと「極秘」であったファイルを、「一般」のファイルとして保存することができてしまい、結果的に、本来「一般」のクリアランスしか持たないユーザに「極秘」の情報が漏洩してしまう。強制アクセス制御では、このような危険性を防止できる。

【００２８】

本実施例の強制アクセス制御におけるファイルアクセスの種類（アクセスタイプという）として、ファイルを開いて内容を読み出すこと、ファイルの情報を書き換えること、新たにファイルを作成すること、を定義する。

ファイルの情報の書き換えにおいて、ファイルを開くかどうかは問わない。すなわち、「書き換えること」ができても「読み出すこと」ができないことがあり得る。つまり、読み取りできない(内容のわからない)ファイルの内容を書き換えることができる権利をもつことがありえる。これによって、クリアランスの低いユーザが高いセキュリティレベルのファイルを書き換える際に、情報が漏れることを防ぐ。

【００２９】

さらに、書き換えについては、ファイルのセキュリティレベルを下げられないようにするという目的を達成するためであれば、元のファイル情報の書き換えを含む書き込みを許してもよいが、クリアランスの低いユーザによって、高いセキュリティレベルの情報を書き換えられることを避けるという観点からは、たとえば追記のみを許すようにしてもよい。

【００３０】

すなわち、ファイル内の情報を減らすようなファイルアクセスを「削除」と呼び、ファイル内の情報を増やすようなファイルアクセスを「追記」と呼ぶとすると、情報の不正な漏洩を防ぐという目的だけであれば、与えられたクリアランスと区画が同じで、かつ、機密度合いが同じか高いセキュリティレベルのファイルに対して、「削除」と「追記」の両方を許すようにしてもよいし、情報の改竄を防ぐという目的からは、「削除」は許さず「追記」を許すようにしてもよい(この場合、たとえば、機密度合いが同じ場合のみ「削除」を許すようにしてもよい)。

【００３１】

ユーザのクリアランスは、セキュリティ管理者がユーザラベル設定プログラム１０２３を利用することによって、たとえば、ホストOS１０１７上で最初にユーザ登録を行うときに設定する。あるいは、当該ユーザが最初に何らかのファイルアクセスを行おうとしたときに、当該ファイルアクセスをフックすることにより、ユーザラベル設定プログラム１０２３を自動的に起動するようにしてもよい。ユーザラベル設定プログラム１０２３の起動が設定されていない場合には、ユーザのクリアランスが設定されていない状態で(すなわち、ゲストOS１０１９側で管理されているユーザリスト１０４１に当該ユーザに関するクリアランスが登録されていない状態で)、ファイルアクセスが発生する可能性があるが、この場合には、システムのセキュリティポリシーにあわせて、たとえば、いかなるファイルアクセスも許さないようにしてもよいし、あるいは、最も低いクリアランス（本実施例の場合は「一般」）が与えられているものとみなしてもよい。

【００３２】

ファイルのセキュリティレベルは、ユーザが当該ファイルを作成したときに、

たとえばユーザが明示的に設定する。設定可能なセキュリティレベルは、当該ユーザのクリアランスによって決まる。すなわち、当該ユーザのクリアランスと同じかそれより高いセキュリティレベルを設定可能である。また、一度設定されたセキュリティレベルは、一般には、後から下げることはできないようにしたほうがセキュリティの観点からは望ましい。但し、利便性向上等の目的で、特別の権限を持つユーザ、例えばセキュリティ管理者、がファイルラベル設定プログラム１０２５を利用することにより、セキュリティレベルを下げられるようにしてもよい。

【００３３】

ユーザのクリアランスを設定するセキュリティ管理者と、ファイルのセキュリティレベルを設定するセキュリティ管理者は同一でもよいし、異なっていてもよい。さらには、それぞれの管理者は信頼できる複数の人であってもよい。たとえば、各区画ごとにユーザのクリアランスおよびファイルのセキュリティレベルを設定するセキュリティ管理者を設けてもよい。

【００３４】

図２を用いて、本実施例におけるファイルアクセス制御の処理フロー(読み出し、書き込み)を詳細に説明する。

ステップ２００１：はじめ

ステップ２００２：アプリケーションプログラム１０２１は、I/0マネージャ１０２７を通じ、ファイルアクセスフックプログラム１０２９に対し、ファイルアクセスを要求

ステップ２００３：ファイルアクセスフックプログラム１０２９は、OS間通信プログラム１０５１を利用し、アクセス制御プログラム１０３５に、ユーザ名、ファイル名、アクセスの種類を送る。ユーザ名は、たとえば、アプリケーションプログラム１０２１に関するプロセスとそのプロセスを実行したユーザ名との対応情報を参照することによって取得できる。この対応情報は、ホストOS自身の持つ機能のひとつであるプロセス管理機能を実現するために使われるもので、ホストOSが保持している。

ステップ２００４：アクセス制御プログラム１０３５は、ユーザリスト１０４１

およびファイルリスト１０４３を参照し、当該ユーザのクリアランス、および、当該ファイルのセキュリティレベルを調べる

ステップ２００５：アクセス制御プログラム１０３５は、正当なアクセスか否かを判定する。正当なアクセスであれば、ステップ２００８へ。不当なアクセスであれば、ステップ２００６へ

ステップ２００６：アクセス制御プログラム１０３５は「アクセス不可」を出力する。この判定結果はファイルフックプログラム１０２９、I/0マネージャ１０２７を通じて、アプリケーションプログラム１０２１に伝えられる

ステップ２００７：おわり（アクセス不可）

ステップ２００８：アクセス制御プログラム１０３５は、ファイルフックプログラム１０２９を通じて、ファイルシステムドライバ１０３１に、ファイル名、アクセスの種類(アクセスタイプ)を送る

ステップ２００９：ファイルシステムドライバ１０３１は、デバイスドライバ１０３３に、対象となるファイル名に応じたハードディスクＡ１０１３内のアドレス、および、アクセスの種類を送る

ステップ２０１０：デバイスドライバ１０３３は、ハードディスクＡ１０１３から読み込み対象となっているファイルを読み出してアプリケーションプログラム１０２１に送る（読み込み時）、あるいは、ハードディスクＡ１０１３にアプリケーションプログラム１０２１が書き込もうとしている情報を書き込む（書き込み時）ステップ２０１１：おわり（アクセス成功）。

【００３５】

図３を用いて、本実施例におけるファイルアクセス制御の処理フロー(ファイル作成)を詳細に説明する。

ステップ３００１：はじめ

ステップ３００２：アプリケーションプログラム１０２１は、I/0マネージャ１０２７を通じ、ファイルアクセスフックプログラム１０２９に対し、ファイルアクセス(ファイル作成)を要求

ステップ３００３：ファイルアクセスフックプログラム１０２９は、アプリケーションプログラム１０２１のユーザに対し、作成するファイルに設定するセキュ

リティレベルの入力を、そのためのユーザインタフェースを用いて、求める
ステップ３００４：ファイルアクセスフックプログラム１０２９は、OS間通信プログラム１０５１を利用し、アクセス制御プログラム１０３５に、ユーザ名、ファイル名、アクセスの種類(ファイル作成)、設定セキュリティレベルを送る。ユーザ名は、ステップ２００３と同様に取得できる。
ステップ３００５：アクセス制御プログラム１０３５は、ユーザリスト１０４１を参照し、当該ユーザのクリアランスを調べる
ステップ３００６：アクセス制御プログラム１０３５は、設定しようとしているセキュリティレベルが正当か否かを判定する。正当であれば、ステップ３００９へ。不当なアクセスであれば、ステップ３００７へ
ステップ３００７：アクセス制御プログラム１０３５は「アクセス不可」を出力する。この判定結果はファイルフックプログラム１０２９、I/0マネージャ１０２７を通じて、アプリケーションプログラム１０２１に伝えられる
ステップ３００８：おわり（アクセス不可）
ステップ３００９：アクセス制御プログラム１０３５は、ゲストOS１０１９管理下のファイルリスト１０４３を更新し、I/0マネージャ１０２７を通じて、ファイルシステムドライバ１０３１に、ファイル名、アクセスの種類(ファイル作成)を送る
ステップ３０１０：ファイルシステムドライバ１０３１は、ファイル名に応じたハードディスクＡ１０１３内のアドレスを割り当て、デバイスドライバ１０３３に、当該アドレスとアクセスの種類を送る
ステップ３０１１：デバイスドライバ１０３３は、ハードディスクＡ１０１３にアプリケーションプログラム１０２１が保存しようとしている情報を書き込む
ステップ３０１２：おわり（アクセス成功）。

【００３６】

図６は、本実施例におけるユーザリスト１０４１の一例である。１列目にファイル名６００１が示されており、２列目に当該ファイルのセキュリティレベル６００３が示されている。

【００３７】

図７は、本実施例におけるファイルリスト１０４３の一例である。１列目にユーザ名７００１が示されており、２列目に当該ユーザのクリアランス７００３が示されている。

【００３８】

図８は、上述の強制アクセス制御の考え方に基づき、本実施例におけるユーザ”researcher”がアクセスすることが許されるファイルのセキュリティレベルを示したものである。２列目から５列目までが、区画（人事部門８００１、８０２１、営業部門８００３、８０２３、企画部門８００５、８０２５、研究部門８００７、８０２７）を、２行目から５行目目がセキュリティレベル（機密８００９、８０２９、極秘８０１１、８０３１、秘密８０１３、８０３３、一般８０１５、８０３５）を示している。

【００３９】

図７に示すように”researcher”に与えられたクリアランスは、「研究部門・極秘」および「企画部門・秘密」であるので、上述の強制アクセス制御の考え方に基づくと、読み取りについては、「研究部門・極秘、秘密、一般」「企画部門・秘密、一般」のファイルにアクセスできる。また、書き込み、または、作成については、「研究部門・機密、極秘」「企画部門・機密、極秘、秘密」のファイルにアクセスできる。したがって、たとえば、ファイル”C:¥home¥user2¥research_report.doc”のセキュリティレベルは、「研究部門・極秘」であるので、読み書き可能であるが、ファイル” C:¥home¥user1¥secret.txt” のセキュリティレベルは、「営業部門・極秘、企画部門・極秘」であるので、書き込みのみ可能である。

【００４０】

アクセス制御プログラム１０３５は、図8に示す内容をルール化、あるいは、テーブル化したものをあらかじめ備え、その内容を参照しつつ、アクセス制御を行う。

図8の表では、特定のユーザがアクセス可能なファイルのセキュリティレベルを示しているが、特定のファイルへのアクセス可能なユーザを示すように作成してもよい。

【００４１】

以上に示したとおり、本実施例は、ホストOS１０１７に、アプリケーションプログラムに優先して働く強制アクセス制御機能を付加するものである。したがって、本実施例によって、ホストOS１０１７上で動作するアプリケーションプログラムにもともと許されていたファイルアクセスが禁止される可能性があるが、当該アプリケーションプログラムを変更する必要はなく、そのまま使用できる。

【００４２】

一例として、ホストOS１０１７管理下で、テキストエディタプログラムを実行させた場合の動作を述べると次のようになる。ここでは、ユーザUのクリアランスが、「研究部門」の「極秘」であるとし、ファイルF1のセキュリティレベルが「研究部門」の「機密」、ファイルF2のセキュリティレベルが「研究部門」の「秘密」であるとする。ユーザUは、テキストエディタプログラムを実行し、当該テキストエディタプログラムの「ファイルを開く」という機能を用い、ファイル名「F1」を指定し、ファイルの内容を読み込もうとした場合、これは、強制アクセス制御によれば許されないアクセスであるので、本実施例に示した機能によりこのアクセスは禁じられ、ユーザには、ファイルを開けない旨、伝えられる(表示される)。

【００４３】

このユーザへの表示は、たとえば、ホストOS１０１７自体が備える機能を利用して行えばよい。すなわち、本実施例によるアクセス制御とは別に、ホストOS１０１７自体が備えるファイルアクセス制御機能において、「読み込み禁止」となっているファイルを読み出そうとしたときの表示を利用すればよい。

同様に、「ファイルを開く」機能によってファイル名「F2」を指定し、ファイルの内容を読み込もうとした場合には、許されるアクセスであるので、ファイルの内容がテキストエディタプログラムに読み込まれる(ユーザから見ると通常のOS上での動作と同じに見える)。

さらにこのファイルを編集したのち、やはりテキストエディタに備えられた「上書き保存」機能を利用して、ファイル名「F2」として上書きしようとすると、これは許されないアクセスであるので、このアクセスは禁じられ、ユーザにはファイルが保存できない旨、伝えられる(表示される)。読み込みの場合と同様に、

このユーザへの表示は、たとえば、ホストOS１０１７自体が備える機能を利用して行えばよい。

【００４４】

本実施例を「追記」のみを許すように設定し、このテキストエディタプログラムが、ファイルを開くことなく追記する、ことに対応しているときには、ユーザUはファイルF1に追記可能である。しかし、一旦ファイルの内容を読み込んでからでないと、編集(この場合は追記)できないテキストエディタプログラムの場合には、ファイルを読み込む(開く)ことが許可されないため、結果的に追記できない。

【００４５】

同一ハードディスク内でのファイル移動のように、ハードディスクＡ１０１３内のデータをメモリA１００５内に読み込むことなく、(パス情報を含むファイル名の書き換えを行うだけ、すなわち、ファイルそのものを移動せず、ファイル名とそのファイルがハードディスク上のどこにかかれているか、を示す対応表の部分だけを書き換えることで実現できる場合がある。このようなファイルアクセスについては、任意のユーザが行えるようにしてもよい。ただし、このとき、ファイルリスト１０４３の整合性を保つために、ファイルリスト１０４３内のファイル名６００１の部分を、当該ファイル移動に合わせて書き換えるようにする。

【００４６】

また、ファイルのコピーについても、コピーの途中で当該ファイルのデータを不正に読み取られる危険性がない場合、たとえば、ハードディスク自体がファイルコピーの機能を持っているような場合には、任意のユーザが行えるようにしてもよい。ただし、このときファイルのセキュリティレベルも同時にコピーする。すなわち、ファイルリスト１０４３内に、当該ファイルに関するエントリを追加して、コピー元と同じセキュリティレベルを設定する必要がある。

また、フロッピーディスクなどの取り外し可能な可搬型記憶媒体に移動またはコピーする場合には、ファイル移動（またはコピー）時に、当該ファイルのセキュリティレベルに合わせて暗号化するようにすればよい。さらに、ファイルリスト１０４３内の当該ファイルに関する情報も、可搬型記憶媒体に書き込むように

してもよい。

これらの処理は、たとえば、アクセス制御プログラム１０３５に機能を追加することによって、実現可能である。

【００４７】

図４を用いて、本実施例におけるユーザラベル設定時における処理フローを説明する。

(ユーザラベル設定プログラム１０２３(ホストOS１０１７上)の動作)

ステップ４００１：はじめ

(ユーザリスト管理プログラム１０３７(ゲストOS１０１９上)の処理)

ステップ４００２：ホストOS１０１７上の起動中のプロセスチェックを行う。信頼できないプロセスが起動している場合は、ステップ４００３へ。そうでなければステップ４００４へ

ステップ４００３：おわり(設定失敗)

ステップ４００４：乱数Rを生成し、ホスト側へ送信する

(ユーザラベル設定プログラム１０２３(ホストOS１０１７上)の動作)

ステップ４００５：設定情報Cu（ユーザ名、当該ユーザのクリアランスレベルなど）の入力を求める

ステップ４００６：セキュリティ管理者のパスワードPの入力を求める

ステップ４００７：乱数Rと設定情報Cuを結合(concatinate)したものを、入力されたパスワードを鍵として暗号化し、ゲスト側へ送信する

(ユーザリスト管理プログラム１０３７(ゲストOS１０１９上)の処理)

ステップ４００８：あらかじめゲストOS１０１９側で管理・保持されていたセキュリティ管理者のパスワードを用い、ゲスト側から送られてきた暗号文を復号する。復号した情報に含まれている乱数が、ステップ４００４で生成した乱数Rと等しくなければ、ステップ４００９へ。等しければ、ステップ４０１０へ

ステップ４００９：おわり(設定失敗)

ステップ４０１０：ユーザリスト１０４１にステップ４００８で復号された設定情報を反映させる

ステップ４０１１：おわり(設定成功)。

【００４８】

図５を用いて、本実施例におけるファイルラベル設定時における処理フローを説明する。

(ファイルラベル設定プログラム１０２５(ホストOS１０１７上)の動作)

ステップ５００１：はじめ

(ファイルリスト管理プログラム１０３９(ゲストOS１０１９上)の処理)

ステップ５００２：ホストOS１０１７上の起動中のプロセスチェックを行う。信頼できないプロセスが起動している場合は、ステップ５００３へ。そうでなければステップ５００４へ

ステップ５００３：おわり(設定失敗)

ステップ５００４：乱数Rを生成し、ホスト側へ送信する

(ファイルラベル設定プログラム１０２５(ホストOS１０１７上)の動作)

ステップ５００５：設定情報Cf（ファイル名、当該ファイルのセキュリティレベルなど）の入力を求める

ステップ５００６：セキュリティ管理者のパスワードPの入力を求める

ステップ５００７：乱数Rと設定情報Cfを結合(concatinate)したものを、入力されたパスワードを鍵として暗号化し、ゲスト側へ送信する

(ファイルリスト管理プログラム１０３９(ゲストOS１０１９上)の処理)

ステップ５００８：あらかじめゲストOS１０１９側で管理・保持されていたセキュリティ管理者のパスワードを用い、ゲスト側から送られてきた暗号文を復号する。復号した情報に含まれている乱数が、ステップ５００４で生成した乱数Rと等しくなければ、ステップ５００９へ。等しければ、ステップ５０１０へ

ステップ５００９：おわり(設定失敗)

ステップ５０１０：ファイルリスト１０４３にステップ５００８で復号された設定情報を反映させる

ステップ５０１１：おわり(設定成功)。

【００４９】

本実施例においては、ユーザラベルまたはファイルのセキュリティレベルの設定を行うときに、ゲストOS１０１９側のユーザリスト管理プログラム１０３７ま

たはファイルリスト管理プログラム１０３９とセキュリティ管理者との間の認証を、チャレンジレスポンス認証(challenge-response identification)と呼ばれる方法を用いておこなっている。さらに、認証と設定情報の受け渡しを同時に行うことにより、認証処理をバイパスする等の不正を防ぎ、かつ、メッセージのやりとりの回数を減らすことができる。

【００５０】

チャレンジレスポンス認証については、

文献[HAC]：Alfred J.Menezes, Paul C. van Oorschot, Scott A. Vanstone, “Handbook of Applied Cryptography”, CRC Press

に詳しい。なお、チャレンジレスポンス認証で利用する、暗号化処理は、たとえばDES(The Data Encryption Standard)

をCBCモード(Cipher-block chaining mode)で利用することによって行えばよい(文献[HAC]参照)。DES以外の共通鍵暗号を用いてもよい。本実施例では、上記文献[HAC]に開示されたチャレンジレスポンス認証以外の認証方法(たとえば公開鍵暗号に基づく方法)を利用することも可能である。

【００５１】

以上のように、本実施例は、ユーザが直接利用する一般的なOSであるホストOS１０１７とは異なるゲストOS１０１９上に、強制アクセス制御機能を設けることにより、ホストOS１０１７上のアプリケーションプログラム１０２１に改変を加えることなく強制アクセス制御機能を実現可能にしつつ、さらに強制アクセス制御機能に対する不正を防止する。

また、ホストOS１０１７の多機能化・高機能化等の改良が行われた際にも、構成を変えることなく強制アクセス制御を可能とする。

さらに、ユーザのクリアランスやファイルのセキュリティレベルの設定にあたっては、(一般にはホストOS１０１７側の管理者とは異なる)セキュリティ管理者の認証を行うことにより、ホストOS１０１７側の管理者による不正の脅威を減らすことができるようになる。

【００５２】

（第２の実施例）

本発明の他の実施の形態を、以下、図を用いて説明する。

【００５３】

前述したように、第１の実施例に開示したシステムが正常に起動した場合、ファイルへの不正アクセスは非常に困難になり、システムを安全に運用することが可能となる。このシステムに対して考えられる更なる脅威として、たとえば、システム起動以前にホストOSのプログラムが記述されたファイルを改竄し、第１の実施例に示したゲストOS側のアクセス制御機能を回避するという不正行為が考えられる。本実施例はこの不正行為を防止する方法を提供する。

【００５４】

図９は、本実施例におけるコンピュータシステムの構成例である。図１と同じであるが、ホストOS１０１７管理下のハードディスクA１０１３内のファイルが暗号化されている点（暗号化されたファイル９００１）、および、ゲストOS１０１９管理下に、暗号プログラム９００３、復号プログラム９００５、および、暗復号化鍵(対称鍵(symmetric key)ともいう)９００７が追加されている点が異なる。

【００５５】

本実施例においては、ファイル書き込み時には、まず、第１の実施例と同様にしてアクセス制御を行い、さらに、正当なアクセスであった場合には、ゲストOS側の暗号プログラム９００３によって暗号化されてから、ホストOS１０１７側のハードディスクＡ１０１３内に保存される。

ファイル読み込み時には、まず、第１の実施例と同様にしてアクセス制御を行い、さらに、正当なアクセスであった場合には、ホストOS１０１７側のハードディスクＡ１０１３に保存された当該ファイル(暗号化されたファイル)はゲストOS側の復号プログラム９００５によって復号されたのち、ホストOS１０１７側のアプリケーションプログラム１０２１に渡される。

【００５６】

この結果、ホストOS側のファイルアクセスが、ゲストOS側のアクセス制御プログラムを利用するようになっている場合には(すなわち、ホストOS側のアクセス制御プログラムをバイパスしていないときには)、ハードディスクA１０１３内の

ファイルは、常に暗号化された状態でありながら、アプリケーションプログラム１０２１は、正当なアクセスである限り、暗号化されていることを意識することなく、透過的に利用可能となる。

【００５７】

一方、たとえば、ホストOSがメモリにロードされるより前に、ホストOSのプログラムが記述されたファイルを改竄することにより、不正なホストOSを起動し、ゲストOS側のアクセス制御機能を回避するという不正行為を行った場合には、読み出されたファイルは暗号化されているため、実際上意味をなさない(情報の流出はおこらない)。したがって、結果的には、ゲストOS、および、当該OS上のアクセス制御機能の起動が保証される。さらには、これにより、たとえハードディスク装置ごと取り外し、別のコンピュータに接続してファイルを解析しようとしても、ファイルが暗号化されているため解析できないという効果もある。

【００５８】

なお、さらにホストOSの不正がないことを、より確実に確認するために、ゲストOS上に、ホストOSの検証機能を設けてもよい。これは、たとえば、ゲストOSの管理下に、あらかじめ、改竄されていないホストOSプログラムのディジタル署名を保持しておき、ホストOSの起動時に、ゲストOS側において、実際に起動されるホストOSプログラムが改竄されていないことを、当該ディジタル署名の検証手続きによって確認する入出力処理監査処理部をゲストOS側に設ければよい。また、入出力処理監査処理部が、ホストOS側のプロセス管理機能を利用し、ホストOS管理で不正なプロセスが実行されていないことを監査してもよい。

【００５９】

もしホストOSプログラムが改竄されている場合には、ゲストOSに指示して、あらゆるファイルアクセスを禁止する、あるいは、ホストOSの実行を止める、などにより、ホストOSの機能を無効化すればよい。このとき、さらにコンピュータに接続されたディスプレイ上への表示、あるいは、コンピュータに接続されたスピーカーを利用した警告音、あるいは、その他の外部装置を利用したメッセージ等により、ホストOSが改竄されているために起動できない旨、ユーザに伝えるようにしてもよい。

【００６０】

図１０を用いて、本実施例におけるファイルアクセス制御フロー（読み出し）を説明する。

ステップ１０００１：はじめ

ステップ１０００２：ステップ２００２～２００４と同様

ステップ１０００３：アクセス制御プログラムは、正当なアクセスか否かを判定する（ステップ２００５と同様）。正当であればステップ１０００６へ。そうでなければステップ１０００４へ

ステップ１０００４：ステップ２００６と同様

ステップ１０００５：おわり（アクセス不可）

ステップ１０００６：ステップ２００８～２００９と同様

ステップ１０００７：デバイスドライバは、ハードディスクから暗号化されたファイルの読み出しを行いファイルフックプログラムに送る

ステップ１０００８：ファイルアクセスフックプログラムは、OS間通信プログラムを利用し、復号プログラムに、暗号化されたファイルを送る

ステップ１０００９：復号プログラムは、ゲストOS側で管理された暗復号化鍵を用いて、暗号化されたファイルを復号し、OS間通信プログラム、ファイルアクセスフックプログラムを通じて、アプリケーションプログラムに送る

ステップ１００１０：おわり（アクセス成功）。

【００６１】

図１１を用いて、本実施例における、ファイルアクセス制御フロー（書き込み）を説明する。

ステップ１１００１：はじめ

ステップ１１００２：ステップ２００２～２００４と同様

ステップ１１００３：アクセス制御プログラムは、正当なアクセスか否かを判定する（ステップ２００５と同様）。正当であればステップ１１００６へ。そうでなければステップ１１００４へ

ステップ１１００４：ステップ２００６と同様

ステップ１１００５：おわり（アクセス不可）

ステップ１１００６：ゲストOS上の暗号プログラムは、ゲストOS側で管理された暗復号化鍵を用いて、アプリケーションプログラム１０２１が書き込もうとしている情報を暗号化する

ステップ１１００７：ステップ２００８～２０１０と同様

ステップ１１００８：おわり（アクセス成功）

ファイル生成時におけるファイルアクセス制御の処理フローは、図１１に示したファイルアクセス制御フロー（書き込み）と基本的に同様であるが、ステップ１１００２が、

ステップ１１００２‘：ステップ３００２～３００５と同様

となる点が異なる。

【００６２】

ステップ１００００９の復号化処理、および、ステップ１１００６の暗号化処理では、たとえばDES(The Data Encryption Standard)をCBCモード(Cipher-block chaining mode)で利用すればよい(文献[HAC]参照)。ただし、CBCモードを利用する場合であっても、最初のブロック長(DESの場合であれば64-bit)分については、同じ平文を同じ鍵で暗号化すると、同じ暗号文となってしまう。これを避けるためには、CBCモードで利用される初期値IVをファイルごとに異なるようにすればよい。たとえば、ステップ１１００６で暗号化をする際に64-bitの乱数を生成し、それをIVとして暗号化時に利用するとともに、このIVをファイルリストに項目を追加して保存する。ステップ１００００９中の復号化処理では、ファイルリストから復号対象ファイルに対応するIVを読み出し復号時に利用すればよい。

【００６３】

本実施例では、ひとつの暗復号化鍵を用いて、暗復号化を行っていたが、複数の暗復号化鍵を用いてもよい。たとえば、ファイルのセキュリティレベル・区画ごとに異なる鍵、あるいは、異なる暗号方法を用いてもよい。本実施例を、このように変更した場合、セキュリティレベルが４段階あり、区画が４区画あるので、全部で１６個の異なる暗復号化鍵を用いることになる。これらの鍵はゲストOS側で管理すればよい。さらには、ファイルのセキュリティレベルに応じて、暗号方法、あるいは、鍵として、安全性が異なるものを用いてもよい。（たとえば、

セキュリティレベルが高いファイルに対しては、長い鍵長の暗号化鍵を用いてもよい)。

【００６４】

また、本実施例で用いる暗復号化鍵は、ハードディスクB１０１５とは異なるハードウェアモジュール（例：ICカードなど）に格納しておくようにしてもよい。この場合、さらに当該ハードウェアモジュール内に、ゲストOS１０１９およびその上で動作する各種プログラム（アクセス制御プログラム１０３５など）が改変されていないことを検証する機能を持たせておき、正当であると確認されたときに限り、暗復号化鍵が利用可能になるように（たとえば、メモリB１００７内にロードされるように）しておくと、ゲストOS側の改ざんという脅威にも対抗できるため、より効果的である。

【００６５】

また、本実施例では、ファイルの内容自体を暗号化していたが、そうではなく、ファイルの管理テーブル部を暗号化するようにしてもよい。ファイルの管理テーブルは、通常ハードディスク内の、ファイルとは別領域に記憶されており、ファイル名と、ハードディスク内のどの部分に当該ファイルが保存されているかを示すアドレス情報との対応が、記述されている。したがって、このアドレス情報部分を暗号化することにより、ディスク内のどこにファイル情報が保存されているか調べることが困難になるため、ファイル情報漏洩の防止に効果がある。また、暗号化する対象となるデータサイズが小さいため、暗復号化にかかる処理量が少なくてすむ、という利点がある。

【００６６】

なお、本実施例でも、第１の実施例と同様、ファイルアクセスフックプログラム１０２９が、I/Oマネージャ１０２７のファイルシステムドライバ１０３１に対するファイルアクセス命令を検知し、そのファイルアクセス命令をファイルシステムドライバ１０３１に渡す前に、ゲストOS１０１９上のアクセス制御プログラム１０３５に渡す例を示した。しかし、これとは異なっていてもよい。たとえば、ファイルアクセスフックプログラム１０２９は、ファイルアクセス（例えば、読み出し）命令を、ファイルシステムドライバ１０３１に渡すのと同時、ある

いは後に、アクセス制御プログラム１０３５に渡すようにし、当該ファイルアクセスが正当なものでなかったときは、ステップ１０００９の復号化処理を行わないようにしてもよい。このようにしても、不当な読み出し処理をしようとすると、アプリケーションプログラムにとっては、暗号化されたままのファイルが読み込まれることになるため、情報の漏洩はおこらない。

【００６７】

ゲストＯＳに対しては、上記第１、２の実施例による強制アクセス制御機能だけでなく、さらに別の機能を追加してもよい。

たとえば、ホストOS側でどのようなファイルアクセスがあったか（アクセス時刻、アクセスしようとしたユーザ名・プログラム名、アクセス対象ファイル名、アクセスの種類、ファイルになされた変更内容など）を記録し、保持する機能（ファイルアクセスログ管理機能）をゲストOS側が備えてもよい。

ゲストOS側で実現することにより、ホストOS、および、ホストOS上で動作するアプリケーションプログラムを改変することなく、ファイルアクセスを記録することができるようになり、さらに、記録されたファイルアクセスログが、ホストOS側から改竄、または、消去される危険性がなくなる。

【００６８】

また、ホストOS上のコンピュータウィルスを検知、あるいは、駆除するコンピュータウィルス対策プログラムをゲストOS側に設けてもよい。さらには、検知したウィルスに関する情報をネットワークを通じて、他のコンピュータとの間で送受信する機能をゲストOSに設けてもよい。

【００６９】

あるいはまた、ホストOS側のアプリケーションプログラムから明示的に実行が要求される機能を実現してもよい。たとえば、ディジタル署名生成機能または公開鍵暗号の復号機能などの暗号機能、および、これらの機能において利用される秘密鍵情報の管理などが挙げられる。さらには、発明者等が特願2000-313123号に開示した、ディジタル署名生成時に、その時点での署名履歴を利用する署名方法、および、署名履歴管理方法を実現してもよい。

この場合は、上述のファイルアクセス制御機能や、ファイルアクセスログ管理

機能などとは異なり、ホストOS側アプリケーションプログラムが、あらかじめ、このゲストOS側で実現された機能に対応している必要がある。ただし、実行にあたっては、上記のユーザラベル設定フローやファイルラベル設定フローと同様の方法によって、固有のユーザ認証を行うことにより、ホストOS側のシステム管理者による不正利用を防ぐことができる。したがって、前述のディジタル署名生成機能や、公開鍵暗号の復号機能のような、各ユーザの秘密鍵情報を利用する機能を実現するためには、特に適している。あるいは、上記のコンピュータウィルス対策プログラムをホストOS側のアプリケーションプログラムから明示的に実行が要求される機能として実現してもよい。

【００７０】

上記各実施例においては、複数OS制御技術を利用し、ユーザが直接利用するホストOS上のファイルアクセス制御を、ホストOSとは異なるゲストOS管理下のソフトウェアが実現していた。しかし、本発明はこれに限定されない。たとえば、ゲストOSとゲストOS上に設けられた機能（処理部）のいずれか一つ以上を、ハードウェアとして実現してもよい。この場合、さらに、このハードウェア内に、ユーザリスト、ファイルリスト、セキュリティ管理者のパスワード、ファイル暗号用の暗号化鍵等を記憶するようにしてもよい。

【００７１】

【発明の効果】本発明によれば、情報処理システムの情報セキュリティに関する安全性を、利用者の利便性を損なうことなく、向上させることが可能となる。

【図面の簡単な説明】

【図１】本発明の第１の実施例を実現する計算機の概略構成図である。

【図２】第１の実施例におけるファイルアクセス(読み出し、書き込み)制御の処理フロー図である。

【図３】第１の実施例におけるファイルアクセス(ファイル作成)制御の処理フロー図である。

【図４】第１の実施例におけるファイルアクセス(読み出し、書き込み)制御の処理フロー図である。

【図５】第１の実施例におけるファイルラベル設定時における処理フロー図であ

る。

【図６】第１の実施例におけるユーザリストユーザリストの一例である。

【図７】第１の実施例におけるファイルリストファイルリストの一例である。

【図８】第１の実施例におけるユーザ”researcher”がアクセスすることが許されるファイルのセキュリティレベルを示したものである。

【図９】第２の実施例を実現する計算機の概略構成図である。

【図１０】第２の実施例におけるファイルアクセス(読み出し)制御の処理フロー図である。

【図１１】第２の実施例におけるファイルアクセス(書き込み)制御の処理フロー図である。

【符号の説明】

１００１：コンピュータ

１００３：CPU

１００５：メモリA

１００７：メモリB

１００９：ディスプレイ

１０１１：キーボード

１０１３：ハードディスクA

１０１５：ハードディスクB

１０１７：ホストOS

１０１９：ゲストOS

１０２１：アプリケーションプログラム

１０２３：ユーザラベル設定プログラム

１０２５：ファイルラベル設定プログラム

１０２７：I/Oマネージャ

１０２９：ファイルアクセスフックプログラム

１０３１：ファイルシステムドライバ

１０３３：デバイスドライバ

１０３５：アクセス制御プログラム

１０３７：ユーザリスト管理プログラム

１０３９：ファイルリスト管理プログラム

１０４１：ユーザリスト

１０４３：ファイルリスト

１０４５：ファイル

１０４７：セキュリティ管理者のパスワード

１０４９：複数OS制御プログラム

１０５１：OS間通信プログラム

９００１：暗号化されたファイル

９００３：暗号プログラム

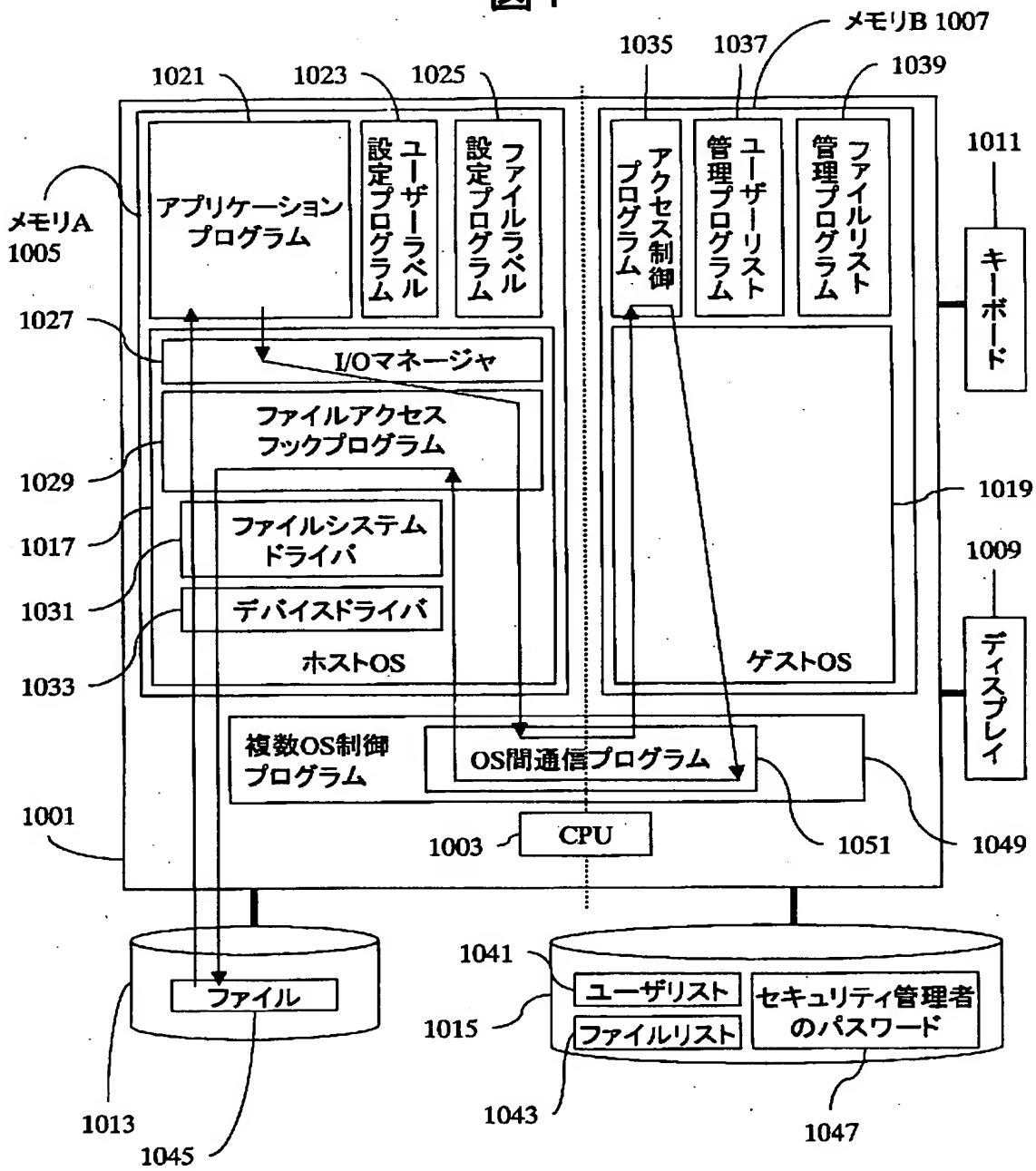
９００５：復号プログラム

９００７：暗号化鍵

【書類名】　図面

【図１】

図１


メモリA
1005
アプリケーション
プログラム
1021
ユーザーラベル
設定プログラム
1023
ファイルラベル
設定プログラム
1025
アクセス制御
プログラム
1035
ユーザーリスト
管理プログラム
1037
メモリB 1007
ファイルリスト
管理プログラム
1039
1011
キーボード
1027
I/Oマネージャ
ファイルアクセス
フックプログラム
1029
1019
1017
ファイルシステム
ドライバ
1009
1031
デバイスドライバ
ディスプレイ
1033
ホストOS
ゲストOS
複数OS制御
プログラム
OS間通信プログラム
1001
1003
CPU
1051
1049
1041
ユーザリスト
セキュリティ管理者
のパスワード
1015
ファイルリスト
ファイル
1043
1047
1013
1045

【図２】

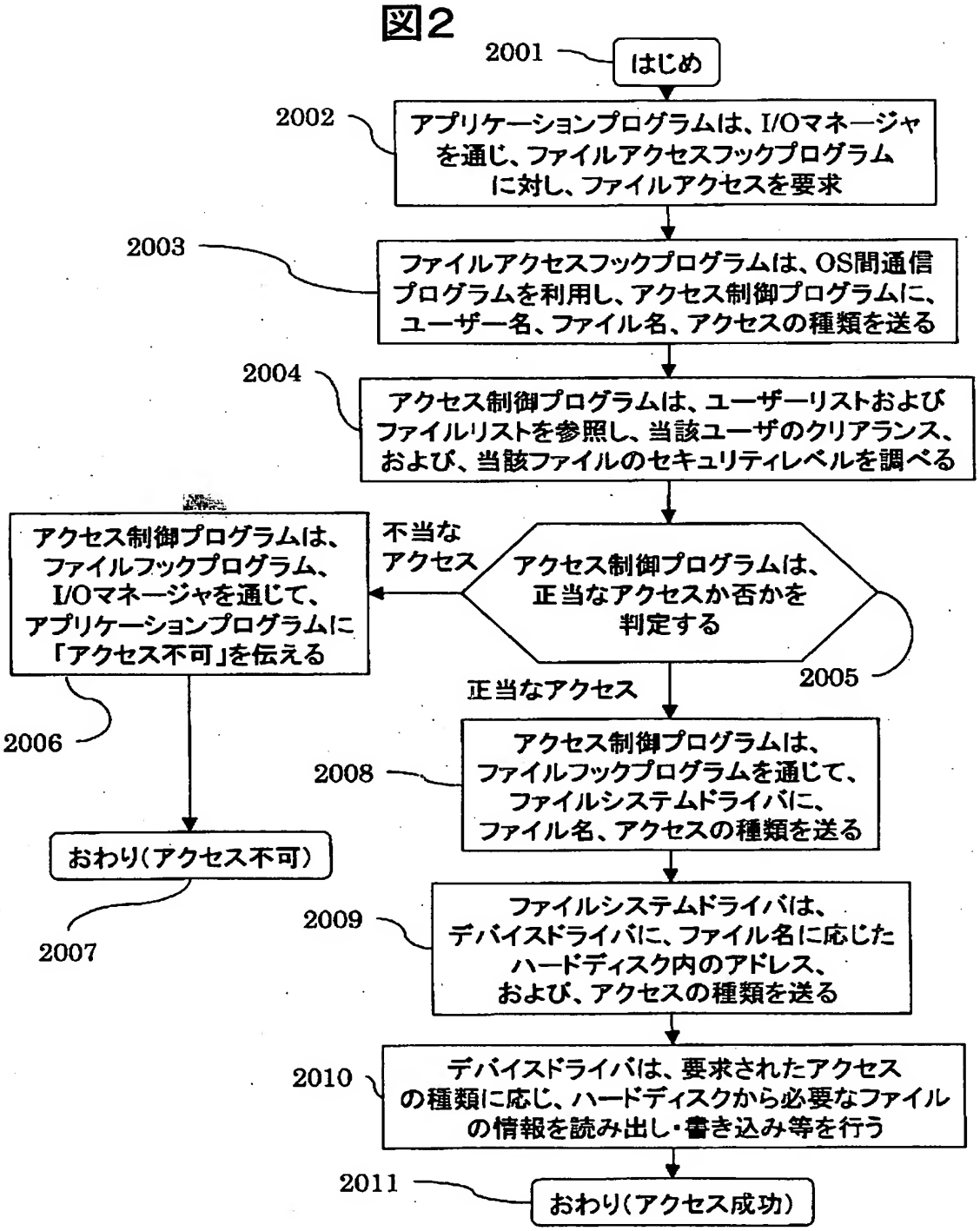
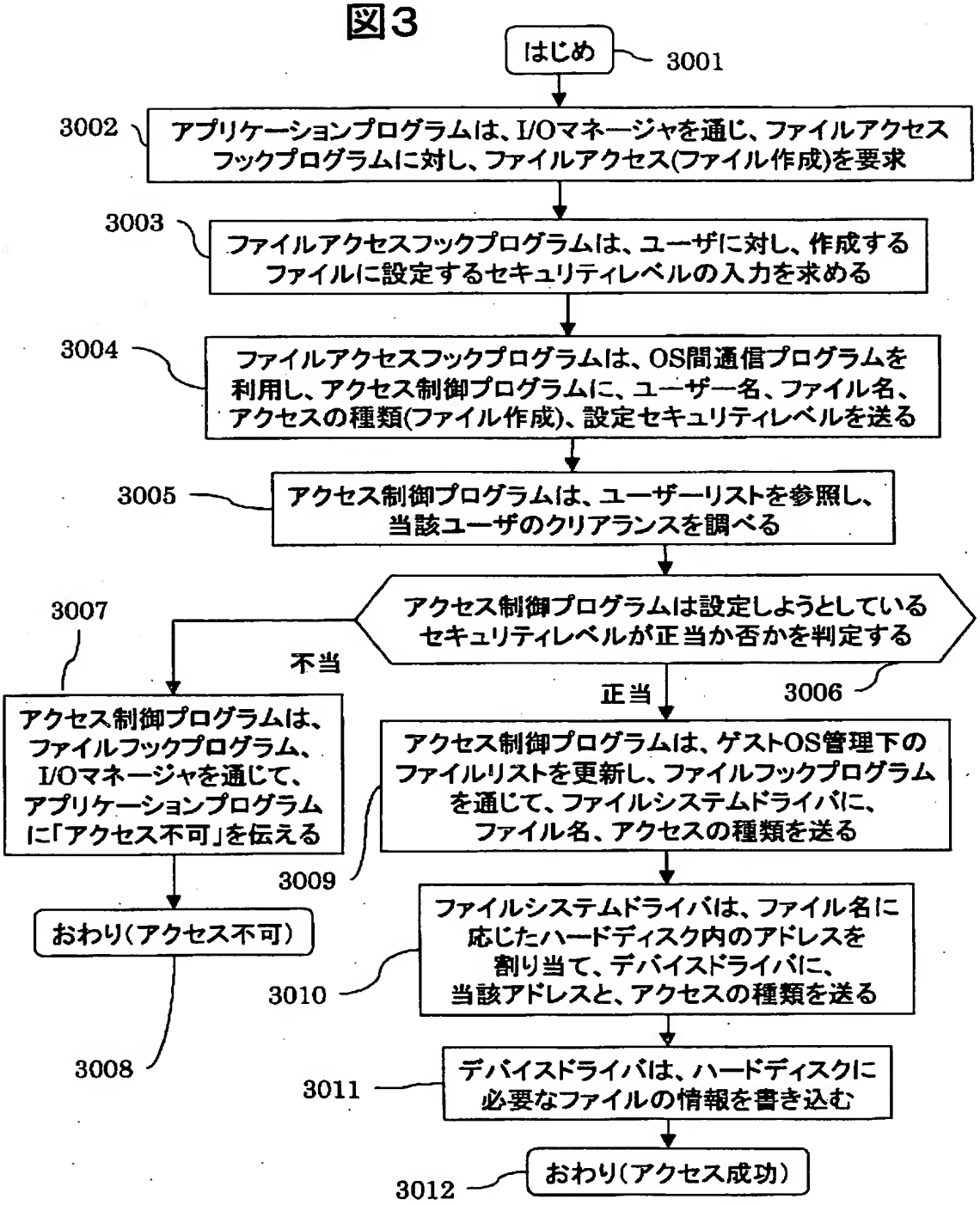


図2
2001
はじめ
2002
アプリケーションプログラムは、I/Oマネージャを通じ、ファイルアクセスフックプログラムに対し、ファイルアクセスを要求
2003
ファイルアクセスフックプログラムは、OS間通信プログラムを利用し、アクセス制御プログラムに、ユーザー名、ファイル名、アクセスの種類を送る
2004
アクセス制御プログラムは、ユーザーリストおよびファイルリストを参照し、当該ユーザのクリアランス、および、当該ファイルのセキュリティレベルを調べる
アクセス制御プログラムは、正当なアクセスか否かを判定する
2005
不当なアクセス
アクセス制御プログラムは、ファイルフックプログラム、I/Oマネージャを通じて、アプリケーションプログラムに「アクセス不可」を伝える
2006
正当なアクセス
2008
アクセス制御プログラムは、ファイルフックプログラムを通じて、ファイルシステムドライバに、ファイル名、アクセスの種類を送る
おわり(アクセス不可)
2007
2009
ファイルシステムドライバは、デバイスドライバに、ファイル名に応じたハードディスク内のアドレス、および、アクセスの種類を送る
2010
デバイスドライバは、要求されたアクセスの種類に応じ、ハードディスクから必要なファイルの情報を読み出し・書き込み等を行う
2011
おわり(アクセス成功)

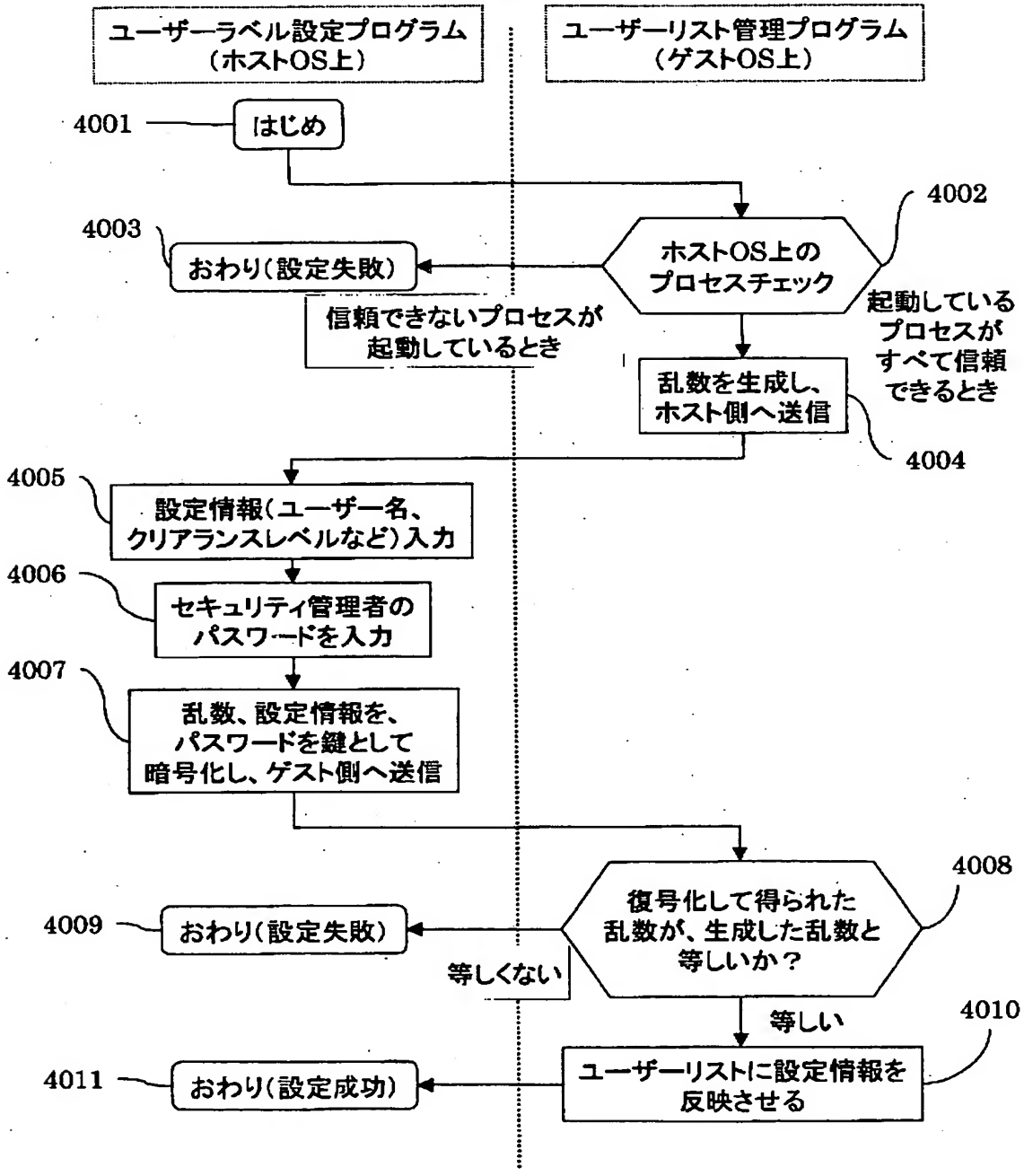
【図３】

図3


はじめ
3001
3002
アプリケーションプログラムは、I/Oマネージャを通じ、ファイルアクセスフックプログラムに対し、ファイルアクセス(ファイル作成)を要求
3003
ファイルアクセスフックプログラムは、ユーザに対し、作成するファイルに設定するセキュリティレベルの入力を求める
3004
ファイルアクセスフックプログラムは、OS間通信プログラムを利用し、アクセス制御プログラムに、ユーザー名、ファイル名、アクセスの種類(ファイル作成)、設定セキュリティレベルを送る
3005
アクセス制御プログラムは、ユーザーリストを参照し、当該ユーザのクリアランスを調べる
アクセス制御プログラムは設定しようとしているセキュリティレベルが正当か否かを判定する
3007
不当
正当
3006
アクセス制御プログラムは、ファイルフックプログラム、I/Oマネージャを通じて、アプリケーションプログラムに「アクセス不可」を伝える
アクセス制御プログラムは、ゲストOS管理下のファイルリストを更新し、ファイルフックプログラムを通じて、ファイルシステムドライバに、ファイル名、アクセスの種類を送る
3009
おわり(アクセス不可)
ファイルシステムドライバは、ファイル名に応じたハードディスク内のアドレスを割り当て、デバイスドライバに、当該アドレスと、アクセスの種類を送る
3010
3008
3011
デバイスドライバは、ハードディスクに必要なファイルの情報を書き込む
おわり(アクセス成功)
3012

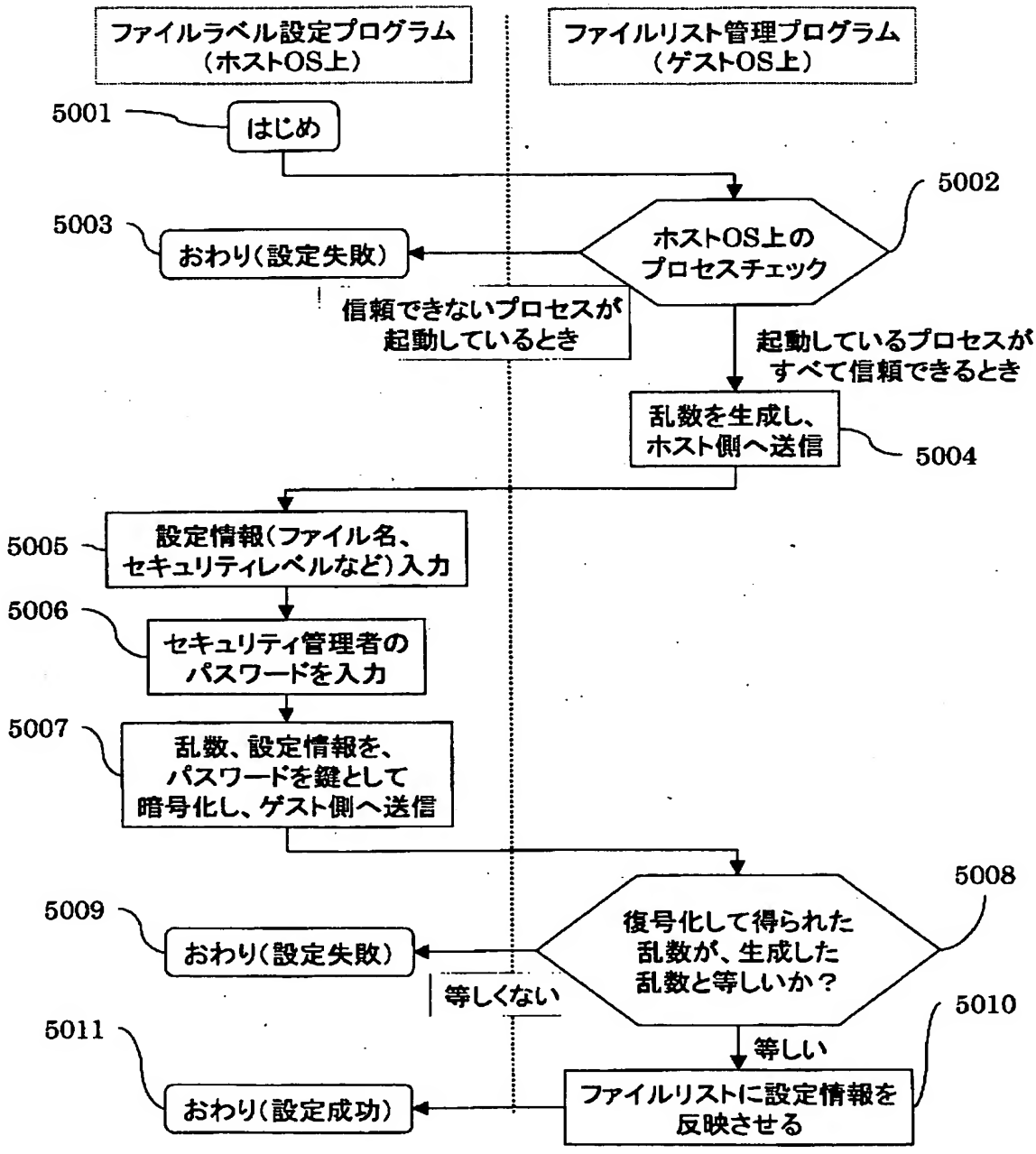
【図４】

図4


ユーザーラベル設定プログラム（ホストOS上）
ユーザーリスト管理プログラム（ゲストOS上）
4001
はじめ
4002
ホストOS上のプロセスチェック
4003
おわり(設定失敗)
信頼できないプロセスが起動しているとき
起動しているプロセスがすべて信頼できるとき
乱数を生成し、ホスト側へ送信
4004
4005
設定情報(ユーザー名、クリアランスレベルなど)入力
4006
セキュリティ管理者のパスワードを入力
4007
乱数、設定情報を、パスワードを鍵として暗号化し、ゲスト側へ送信
4008
復号化して得られた乱数が、生成した乱数と等しいか？
4009
おわり(設定失敗)
等しくない
等しい
4010
ユーザーリストに設定情報を反映させる
4011
おわり(設定成功)

【図５】

図５


ファイルラベル設定プログラム
（ホストOS上）
ファイルリスト管理プログラム
（ゲストOS上）
5001
はじめ
5002
ホストOS上の
プロセスチェック
5003
おわり（設定失敗）
信頼できないプロセスが
起動しているとき
起動しているプロセスが
すべて信頼できるとき
乱数を生成し、
ホスト側へ送信
5004
5005
設定情報（ファイル名、
セキュリティレベルなど）入力
5006
セキュリティ管理者の
パスワードを入力
5007
乱数、設定情報を、
パスワードを鍵として
暗号化し、ゲスト側へ送信
5008
復号化して得られた
乱数が、生成した
乱数と等しいか？
5009
おわり（設定失敗）
等しくない
5010
等しい
5011
おわり（設定成功）
ファイルリストに設定情報を
反映させる

【図６】

図6

6001 ファイル名	6003 セキュリティレベル
C:¥home¥user1¥secret.txt	営業部門・極秘、企画部門・極秘
C:¥home¥user2¥research_report.doc	研究部門・極秘
D:¥public_data¥public.bmp	人事部門・一般、営業部門・一般、 企画部門・一般、研究部門・一般
D:¥secret_data¥top_secret.doc	人事部門・機密
・・・	・・・

【図７】

図7

7001 ユーザ名	7003 クリアランス
researcher	研究部門・極秘、企画部門・秘密
guest	人事部門・一般、営業部門・一般、 企画部門・一般、研究部門・一般
president	人事部門・機密、営業部門・機密、 企画部門・機密、研究部門・極秘
personnel_director	人事部門・機密
・・・	・・・

【図８】

図8

“researcher”が読み取り可能な
ファイルのセキュリティレベル

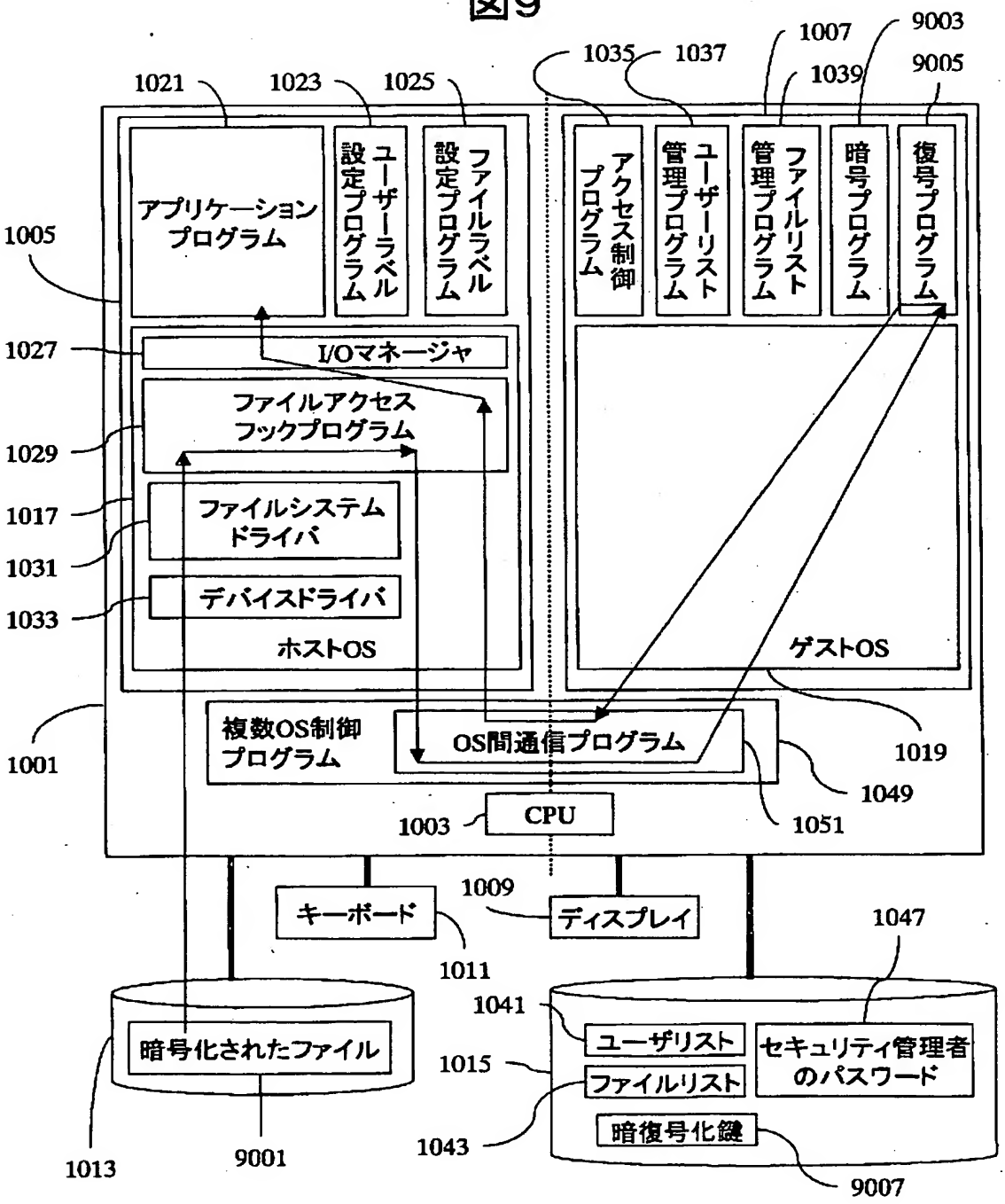
	8001 人事部門	8003 営業部門	8005 企画部門	8007 研究部門
8009 機密				
8011 極秘				○
8013 秘密			○	○
8015 一般			○	○

“researcher”が書き込み、または、
作成可能なファイルのセキュリティレベル

	8021 人事部門	8023 営業部門	8025 企画部門	8027 研究部門
8029 機密			○	○
8031 極秘			○	○
8033 秘密			○	
8035 一般				

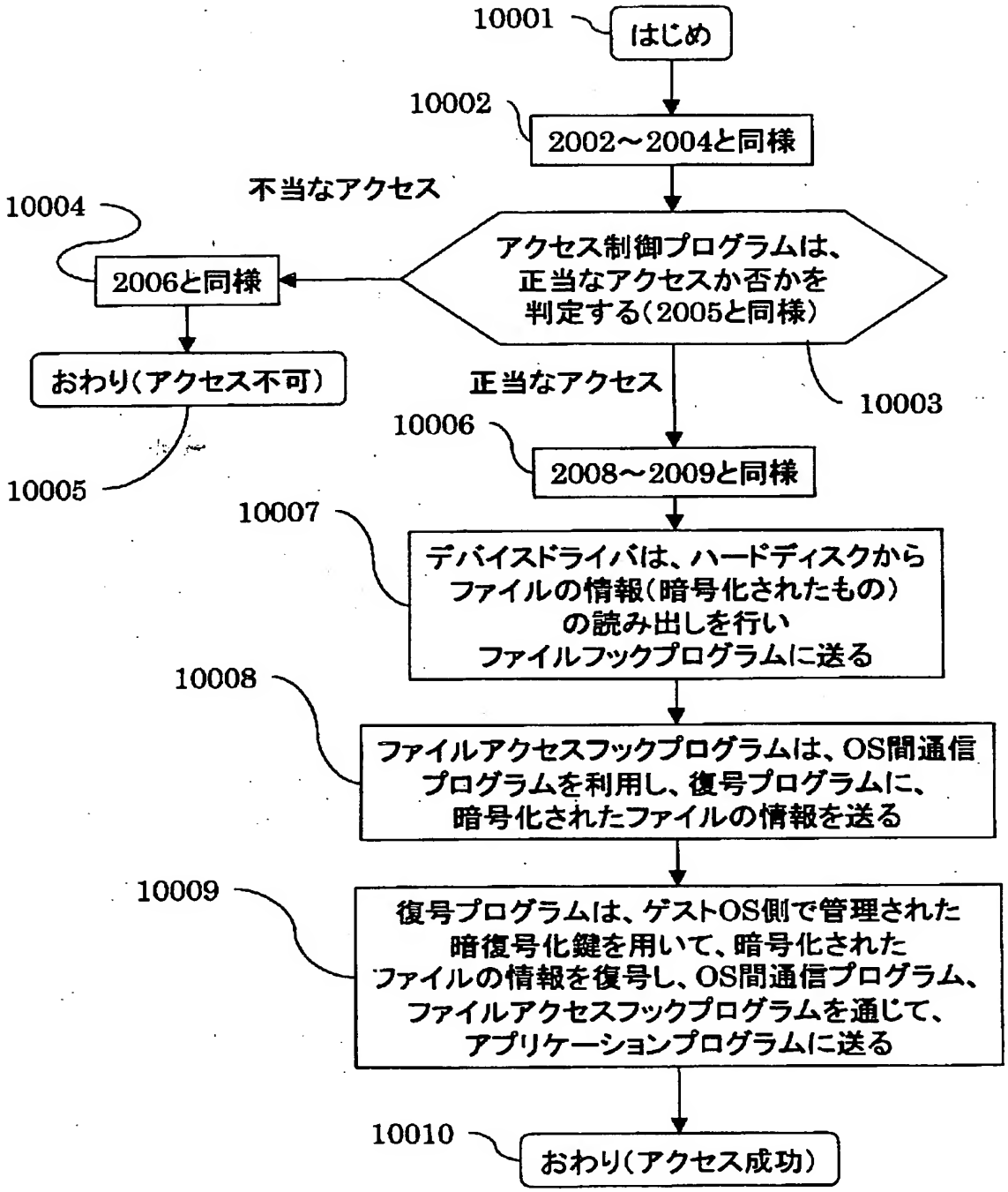
【図9】

図9


1021
1023
1025
1035
1037
1007
1039
9003
9005
アプリケーションプログラム
ユーザーラベル設定プログラム
ファイルラベル設定プログラム
アクセス制御プログラム
ユーザーリスト管理プログラム
ファイルリスト管理プログラム
暗号プログラム
復号プログラム
1005
1027
I/Oマネージャ
ファイルアクセスフックプログラム
1029
1017
ファイルシステムドライバ
1031
1033
デバイスドライバ
ホストOS
ゲストOS
1001
複数OS制御プログラム
OS間通信プログラム
1019
1049
1051
1003
CPU
1009
キーボード
ディスプレイ
1047
1011
1041
暗号化されたファイル
ユーザリスト
セキュリティ管理者のパスワード
1015
ファイルリスト
1043
暗復号化鍵
1013
9001
9007

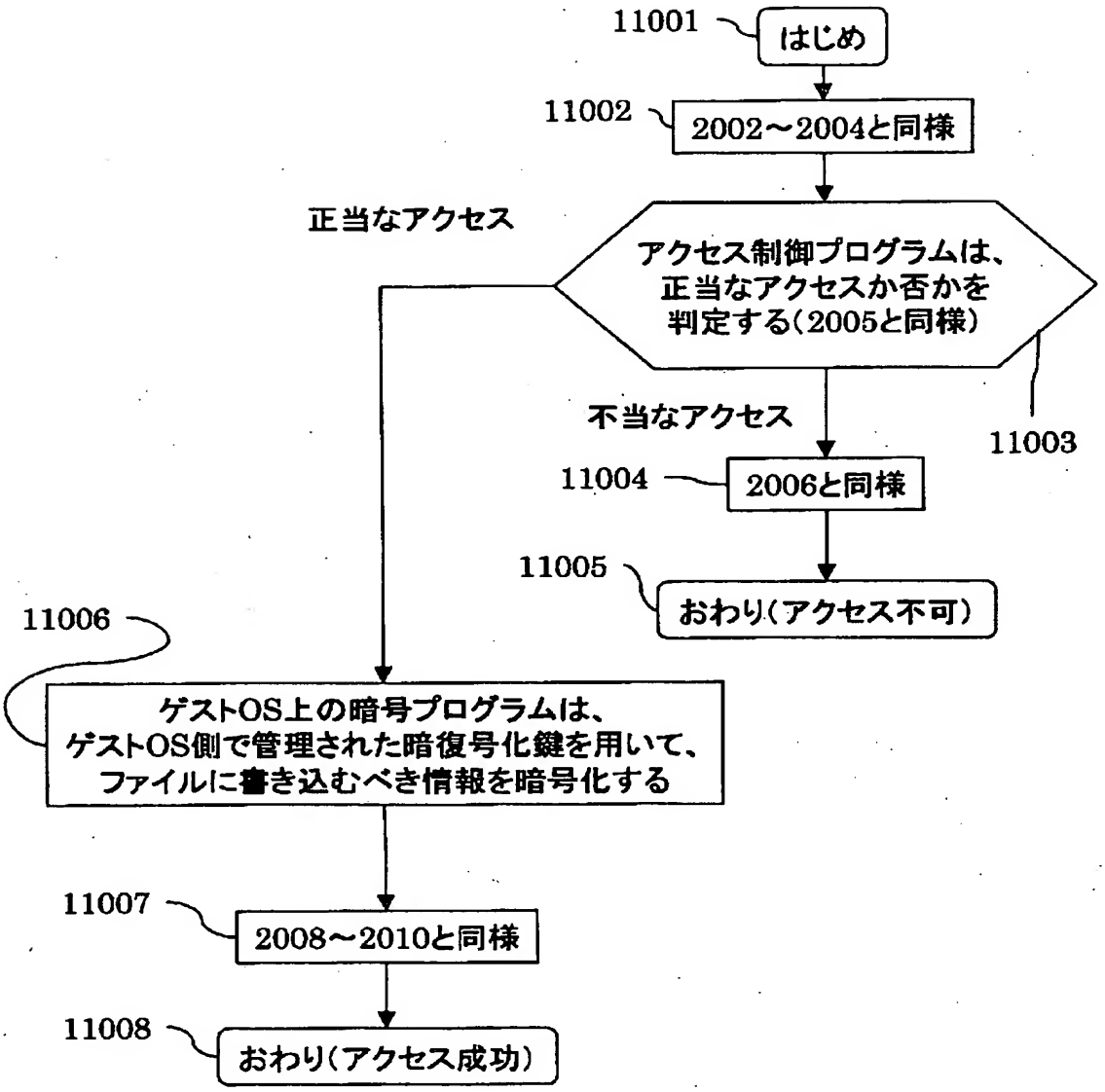
【図１０】

図10


10001
はじめ
10002
2002～2004と同様
不当なアクセス
10004
2006と同様
アクセス制御プログラムは、
正当なアクセスか否かを
判定する（2005と同様）
10003
おわり（アクセス不可）
正当なアクセス
10005
10006
2008～2009と同様
10007
デバイスドライバは、ハードディスクから
ファイルの情報（暗号化されたもの）
の読み出しを行い
ファイルフックプログラムに送る
10008
ファイルアクセスフックプログラムは、OS間通信
プログラムを利用し、復号プログラムに、
暗号化されたファイルの情報を送る
10009
復号プログラムは、ゲストOS側で管理された
暗復号化鍵を用いて、暗号化された
ファイルの情報を復号し、OS間通信プログラム、
ファイルアクセスフックプログラムを通じて、
アプリケーションプログラムに送る
10010
おわり（アクセス成功）

【図１１】

図11


11001
はじめ
11002
2002～2004と同様
正当なアクセス
アクセス制御プログラムは、
正当なアクセスか否かを
判定する（2005と同様）
11003
不当なアクセス
11004
2006と同様
11005
おわり（アクセス不可）
11006
ゲストOS上の暗号プログラムは、
ゲストOS側で管理された暗復号化鍵を用いて、
ファイルに書き込むべき情報を暗号化する
11007
2008～2010と同様
11008
おわり（アクセス成功）

【書類名】　要約書

【要約】

【課題】　ファイルへの不正アクセスを防ぐ仕組みを任意アクセス制御に基づくアクセス制御機能を備えたOSに依存している情報処理システムに対し、よりコストがかからず、安全性を高められる技術を提供する。

【解決手段】　情報処理システム内にシステム管理者とは異なる特定のユーザだけが利用できる機能・領域を設けるための方法、および、装置を提供し、さらに、当該領域にアクセス制御機能を設けることにより、アクセス制御機能に対する不正を防止。

【選択図】　図１

認定・付加情報

特許出願の番号	特願２００１－１２５２２４
受付番号	５０１００５９６１２５
書類名	特許願
担当官	第七担当上席　　００９６
作成日	平成１３年　４月２５日

＜認定情報・付加情報＞

【提出日】　平成13年 4月24日

次頁無

出証特２００２－３００７５８９

出　願　人　履　歴　情　報

識別番号　　〔０００００５１０８〕

１．変更年月日　　１９９０年　８月３１日
［変更理由］　新規登録
住　所　東京都千代田区神田駿河台４丁目６番地
氏　名　株式会社日立製作所